

FILEID**FRMHDR

M 11

FFFFFFFF F RRRRRRRR MM MM HH HH DDDDDDDD RRRRRRRR
FFFFFFFF F RRRRRRRR MM MM HH HH DDDDDDDD RRRRRRRR
FF RR RR MMMM MMMM HH HH DD DD RR RR
FF RR RR MMMM MMMM HH HH DD DD RR RR
FF RR RR MM MM MM HH HH DD DD RR RR
FF RR RR MM MM MM HH HH DD DD RR RR
FFFFFFFF F RRRRRRRR MM MM HHHHHHHHHH DD DD RRRRRRRR
FFFFFFFF F RRRRRRRR MM MM HHHHHHHHHH DD DD RRRRRRRR
FF RR RR MM MM HH HH DD DD RR RR RR
FF RR RR MM MM HH HH DD DD RR RR RR
FF RR RR MM MM HH HH DD DD RR RR RR
FF RR RR MM MM HH HH DDDDDDDD RR RR RR
FF RR RR MM MM HH HH DDDDDDDD RR RR RR

....
....
....
....

LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS
LLLLLLLL L IIIII SSSSSSSS
LLLLLLLL L IIIII SSSSSSSS

FR
VC

```
1 0001 0 MODULE FRMHDR (LANGUAGE (BLISS32) ,
2 0002 0 IDENT = 'V04-000'
3 0003 0 )
4 0004 1 BEGIN
5 0005 1 ****
6 0006 1 ****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 ****
28 0028 1 ++
29 0029 1 +
30 0030 1 |
31 0031 1 FACILITY: MTAACP
32 0032 1 |
33 0033 1 ABSTRACT:
34 0034 1 |
35 0035 1 This module formats HDR1, HDR2, HDR3, and HDR4.
36 0036 1 |
37 0037 1 ENVIRONMENT:
38 0038 1 |
39 0039 1 Starlet operating system, including privileged system services
40 0040 1 and internal exec routines.
41 0041 1 |
42 0042 1 --
43 0043 1 |
44 0044 1 |
45 0045 1 |
46 0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 2-JUN-77 14:35
47 0047 1 |
48 0048 1 MODIFIED BY:
49 0049 1 |
50 0050 1 V03-003 MMD0301 Meg Dumont, 20-Jun-1984 11:26
51 0051 1 Fix to default HDR4 file extension to ASCII zeros instead of
52 0052 1 decimal zeros
53 0053 1 |
54 0054 1 V03-002 MMD0279 Meg Dumont, 23-Mar-1984 10:25
55 0055 1 Fix long file name support such that for ANSI version
56 0056 1 3 volumes it converts the exentstion length to
57 0057 1 ASCII characters before writing it to the label.
```

: 58 0058 1 |
: 59 0059 1 |
: 60 0060 1 |
: 61 0061 1 |
: 62 0062 1 |
: 63 0063 1 |
: 64 0064 1 |
: 65 0065 1 |
: 66 0066 1 |
: 67 0067 1 |
: 68 0068 1 |
: 69 0069 1 |
: 70 0070 1 |
: 71 0071 1 |
: 72 0072 1 |
: 73 0073 1 |
: 74 0074 1 |
: 75 0075 1 |
: 76 0076 1 |
: 77 0077 1 |
: 78 0078 1 |
: 79 0079 1 |
: 80 0080 1 |
: 81 0081 1 |
: 82 0082 1 |
: 83 0083 1 |
: 84 0084 1 |
: 85 0085 1 |
: 86 0086 1 |
: 87 0087 1 |
: 88 0088 1 |
: 89 0089 1 |
: 90 0090 1 |
: 91 0091 1 |
: 92 0092 1 |
: 93 0093 1 |
: 94 0094 1 |
: 95 0095 1 |
: 96 0096 1 |
: 97 0097 1 |
: 98 0098 1 |
: 99 0099 1 |**
: 100 0100 1 |
: 101 0101 1 LIBRARY 'SY\$LIBRARY:LIB.L32';
: 102 0102 1 |
: 103 0103 1 REQUIRE 'SRC\$:MTADEF.B32';
: 104 0487 1 |
: 105 0488 1 FORWARD ROUTINE
: 106 0489 1 FORMAT_HDRS : COMMON_CALL NOVALUE: ! format headers

V03-001 MMD0160 Meg Dumont, 26-Apr-1983 9:31
Add long file name support include: 1) Change FORMAT_FILE_NAME
to understand that VMS long file names are split between
the HDR1 and HDR4 labels. 2) Change FORMAT_HDRS to format the
HDR4. Added support for interchange qualifier.

V02-012 DMW00069 David Michael Walp 11-Jan-1981
Added support of ANSI "a" 17 character filename thru
QIO filename parameter

V02-011 DMW00064 David Michael Walp 6-Jan-1981
Return VMS file spec created by ASCNAME without quotes

V02-010 DMW00063 David Michael Walp 18-Dec-1981
Finished ANSI "a" 17 character file name support

V02-009 DMW00053 David Michael Walp 10-Nov-1981
Return if ANSI resultant name, return it minus trailing
spaces.

V02-008 DMW00043 David Michael Walp 27-Oct-1981
Added ANSI "a" 17 character file name support

V02-007 DMW00016 David Michael Walp 20-May-1981
Get the File Set Id from the MVL rather than 1st volume
mounted label in the VCB.

V02-006 DMW00008 David Michael Walp 23-Jan-1981
Added check for "%" wild card, needed because of expanded
wild card support. Also code commented out for support of
HDR2 attributes.

V02-005 REFORMAT Maria del C. Nasr 30-Jun-1980

A0004 MCN0008 Maria del C. Nasr 22-Feb-1980 16:29
Temporary support of RMS attributes in HDR2

A0003 MCN0003 Maria del C. Nasr 28-Sep-79 10:23
Add HDR3 processing

```
: 108 M 0490 1 MACRO RAD50_TO_VMS ( STRING, VERSION, OUT_INDEX ) =
: 109 M 0491 1
: 110 M 0492 1 ++
: 111 M 0493 1
: 112 M 0494 1 FUNCTIONAL DESCRIPTION:
: 113 M 0495 1 This routine converts a RAD-50 file name block into the
: 114 M 0496 1 equivalent VMS format name. Long file names are not supported
: 115 M 0497 1 in RAD50 mode.
: 116 M 0498 1
: 117 M 0499 1 CALLING SEQUENCE:
: 118 M 0500 1 RAD50_TO_VMS ( ARG1, ARG2, ARG3 )
: 119 M 0501 1
: 120 M 0502 1 INPUT PARAMETERS:
: 121 M 0503 1 none
: 122 M 0504 1
: 123 M 0505 1 IMPLICIT INPUTS:
: 124 M 0506 1 NMBLOCK - the Radix 50 name block
: 125 M 0507 1
: 126 M 0508 1 OUTPUT PARAMETERS:
: 127 M 0509 1 ARG1 - buffer for file name string
: 128 M 0510 1 ARG2 - word to receive version number
: 129 M 0511 1 ARG3 - size of filename string
: 130 M 0512 1
: 131 M 0513 1 IMPLICIT OUTPUTS:
: 132 M 0514 1 none
: 133 M 0515 1
: 134 M 0516 1 SIDE EFFECTS:
: 135 M 0517 1 none
: 136 M 0518 1
: 137 M 0519 1 --
: 138 M 0520 1
: 139 M 0521 1 BEGIN
: 140 M 0522 1
: 141 M 0523 1 EXTERNAL
: 142 M 0524 1 NMBLOCK : VECTOR [ , WORD ]; ! the rad50 name block
: 143 M 0525 1
: 144 M 0526 1 MAP
: 145 M 0527 1 STRING : VECTOR [ , BYTE ]; ! string buffer arg
: 146 M 0528 1 VERSION : WORD; ! version number arg
: 147 M 0529 1 OUT_INDEX : LONG; ! file name size arg
: 148 M 0530 1
: 149 M 0531 1 LOCAL
: 150 M 0532 1 CHARS : VECTOR [ 3, BYTE ]; ! holding place for characters
: 151 M 0533 1
: 152 M 0534 1
: 153 M 0535 1 ! Set up the index. Then start up the outer loop, which iterates
: 154 M 0536 1 over name and type fields.
: 155 M 0537 1
: 156 M 0538 1 OUT_INDEX = 0;
: 157 M 0539 1
: 158 M 0540 1 INCR K FROM 0 TO 3 BY 3 DO
: 159 M 0541 1 BEGIN
: 160 M 0542 1
: 161 M 0543 1 ! The next loop iterates over the RAD-50 words in the name block.
: 162 M 0544 1 There are 3 words for name, 1 for type. Expand each word into
: 163 M 0545 1 the 3 RAD-50 characters.
: 164 M 0546 1
```



```
: 205 M 0586 1 MACRO FORMAT_FILE_NAME =
: 206 M 0587 1
: 207 M 0588 1 ++
: 208 M 0589 1
: 209 M 0590 1 FUNCTIONAL DESCRIPTION:
: 210 M 0591 1 formats the File's Name, Type and Version to placed into the header
: 211 M 0592 1
: 212 M 0593 1 CALLING SEQUENCE:
: 213 M 0594 1 FORMAT_FILE_NAME
: 214 M 0595 1
: 215 M 0596 1 INPUT PARAMETERS:
: 216 M 0597 1 none
: 217 M 0598 1
: 218 M 0599 1 IMPLICIT INPUTS:
: 219 M 0600 1 none
: 220 M 0601 1
: 221 M 0602 1 OUTPUT PARAMETERS:
: 222 M 0603 1 none
: 223 M 0604 1
: 224 M 0605 1 IMPLICIT OUTPUTS:
: 225 M 0606 1 file name is formatted in the HDR1 and HDR4 labels
: 226 M 0607 1
: 227 M 0608 1 SIDE EFFECTS:
: 228 M 0609 1 none
: 229 M 0610 1
: 230 M 0611 1 --
: 231 M 0612 1
: 232 M 0613 1 BEGIN
: 233 M 0614 1
: 234 M 0615 1 EXTERNAL ROUTINE
: 235 M 0616 1 CALC_TAPE_VER, | turn VMS version number
: 236 M 0617 1 PARSE_NAME_TYPE, | into ANSI generation nums
: 237 M 0618 1 PARSE_QUOTED_NAME: COMMON_CALL_NOVALUE, | parse file name string
: 238 M 0619 1 RESULTANT_STRING: COMMON_CALL_NOVALUE, | parse a spec in quotes
: 239 M 0620 1 STRIP_VERSION : COMMON_CALL, | return resultant string
: 240 M 0621 1 SYSSFAO : ADDRESSING_MODE ( ABSOLUTE ); ! format generation num | strip version from file spec
: 241 M 0622 1
: 242 M 0623 1 EXTERNAL
: 243 M 0624 1 ANSI_NAME_SIZE : SIGNED_BYTE, | size of the ANSI file name
: 244 M 0625 1 HDR1 : REF_BBLOCK, | point to the HDR1
: 245 M 0626 1 HDR4 : REF_BBLOCK, | point to the HDR4
: 246 M 0627 1 IO_PACKET : REF_BBLOCK, | pointer to current IRP
: 247 M 0628 1 LOCAL_FIB : BBLOCK; | copy of users File Info Blk
: 248 M 0629 1
: 249 M 0630 1 LOCAL
: 250 M 0631 1 ABD : REF_BBLOCKVECTOR [ , ABD$C_LENGTH ], | pointer to ACP buffer desc
: 251 M 0632 1 DESCRIPT : VECTOR [ 2, LONG ], | general purpose descriptor
: 252 M 0633 1 FILE_SPEC_PTR : LONG, | point to file name buffer
: 253 M 0634 1 FILE_SPEC_LEN : LONG, | length of file name buffer
: 254 M 0635 1 NAME_STRING : VECTOR [ FILE_SPEC_MAX, BYTE ], | buff to hold converted RAD50
: 255 M 0636 1
: 256 M 0637 1 FILE_ID : VECTOR [FILE_SPEC_MAX, BYTE], ! hole FILE ID
: 257 M 0638 1 GEN_NUM_VER : VECTOR [ 2, LONG ], | ANSI version numbers
: 258 M 0639 1 QUOTED_NAME : BITVECTOR [ 1 ], | was the spec passed in quotes
: 259 M 0640 1 VERSION : WORD; | VMS version number
: 260 M 0641 1
: 261 M 0642 1
```

```
: 262 M 0643 1
: 263 M 0644 1 | which filename should be used
: 264 M 0645 1 | get the filename from name block if not specified as attribute
: 265 M 0646 1
: 266 M 0647 1 ABD = .BBLOCK [ .IO PACKET [ IRPSL_SVAPTE ], AIBSL_DESCRIPTOR ];
: 267 M 0648 1 IF .ABD [ ABDSC_NAME, ABDSW_COUNT ] EQLU 0
: 268 M 0649 1 THEN
: 269 M 0650 1 BEGIN
: 270 M 0651 1 RAD50_TO_VMS ( NAME_STRING, VERSION, FILE_SPEC_LEN );
: 271 M 0652 1 FILE_SPEC_PTR = NAME_STRING;
: 272 M 0653 1 QUOTED_NAME [ 0 ] = FALSE;
: 273 M 0654 1 END
: 274 M 0655 1 ELSE
: 275 M 0656 1 BEGIN
: 276 M 0657 1 FILE_SPEC_LEN = .ABD [ ABDSC_NAME, ABDSW_COUNT ];
: 277 M 0658 1 FILE_SPEC_PTR = .ABD [ ABDSC_NAME, ABDSW_TEXT ] +
: 278 M 0659 1 ABD [ ABDSC_NAME, ABDSW_TEXT ] + 1;
: 279 M 0660 1
: 280 M 0661 1 ! do not allow wild cards in the version field
: 281 M 0662 1 !
: 282 M 0663 1 VERSION = STRIP_VERSION ( FILE_SPEC_LEN,
: 283 M 0664 1 FILE_SPEC_PTR,
: 284 M 0665 1 FALSE,
: 285 M 0666 1 QUOTED_NAME [ 0 ] );
: 286 M 0667 1 END;
: 287 M 0668 1
: 288 M 0669 1 ! check that it is not too large
: 289 M 0670 1
: 290 M 0671 1 IF .VERSION GTRU 32767 THEN ERR_EXIT ( SSS_BADFILEVER );
: 291 M 0672 1
: 292 M 0673 1 ! Space fill the temporary FILE_ID field
: 293 M 0674 1 CHSFILL(' ',FILE_SPEC_MAX,FILE_ID);
: 294 M 0675 1
: 295 M 0676 1
: 296 M 0677 1 ! parse the file name if it is a VMS file spec and place into HDR1
: 297 M 0678 1
: 298 M 0679 1 IF .ANSI_NAME_SIZE LSS 0
: 299 M 0680 1 THEN
: 300 M 0681 1 BEGIN
: 301 M 0682 1 DESCRIPT [ 0 ] = FILE_SPEC_MAX;
: 302 M 0683 1 DESCRIPT [ 1 ] = FILE_ID[0];
: 303 M 0684 1
: 304 M 0685 1 ! call the correct parse routine
: 305 M 0686 1
: 306 M 0687 1 IF .QUOTED_NAME [ 0 ]
: 307 M 0688 1 THEN
: 308 M 0689 1 BEGIN
: 309 M 0690 1 PARSE_QUOTED_NAME ( .FILE_SPEC_LEN,
: 310 M 0691 1 FILE_SPEC_PTR,
: 311 M 0692 1 DESCRIPT );
: 312 M 0693 1
: 313 M 0694 1 ! set a dummy value so it is tested to see if it is VMS spec
: 314 M 0695 1
: 315 M 0696 1 ANSI_NAME_SIZE = 1;
: 316 M 0697 1 END
: 317 M 0698 1 ELSE
: 318 M 0699 1 BEGIN
```

```
319 M 0700 1 IF NOT PARSE_NAME_TYPE ( FALSE, ! no wild cards allowed
320 M 0701 1 .FILE_SPEC_LEN,
321 M 0702 1 .FILE_SPEC_PTR,
322 M 0703 1 DESCRIPT )
323 M 0704 1 THEN ERR_EXIT ( SSS_BADFILENAME );
324 M 0705 1 END;
325 M 0706 1 END;
326 M 0707 1
327 M 0708 1 ! test if the file spec give to us by ATR$_ASCNAME or in quotes is a VMS
328 M 0709 1 ! spec so we do not return it in quotes. A size of zero will not work
329 M 0710 1 ! cause you need at least a ""
330 M 0711 1
331 M 0712 1 IF .ANSI_NAME_SIZE GTR 0
332 M 0713 1 THEN
333 M 0714 1 BEGIN
334 M 0715 1 EXTERNAL WORK AREA; ! address of general work area
335 M 0716 1 DESCRIPT [ 0 ] = FILE_SPEC_MAX;
336 M 0717 1 DESCRIPT [ 1 ] = WORK_AREA;
337 M 0718 1 IF PARSE_NAME_TYPE ( FALSE, ! no wild cards allowed
338 M 0719 1 FILE_SPEC_MAX,
339 M 0720 1 FILE_ID[0],
340 M 0721 1 DESCRIPT )
341 M 0722 1 THEN ANSI_NAME_SIZE = -1;
342 M 0723 1 END;
343 M 0724 1
344 M 0725 1 ! Fill in the HDR1 FILE ID field and the HDR4 label.
345 M 0726 1
346 M 0727 1 CHSMOVE (HD1$S_FILEID, FILE_ID, HDR1[HD1$T_FILEID]);
347 M 0728 1 CHSMOVE (HD4$S_FILEID_EXT, FILE_ID[HD1$S_FILEID], HDR4[HD4$T_FILEID_EXT]);
348 M 0729 1
349 M 0730 1 ! Check the length of the file name. If the file name will fit in
350 M 0731 1 the HDR1 FILE ID then set the HDR4 length to zero. Else set up
351 M 0732 1 the lengths such that the HDR1 FILE ID is filled with the name
352 M 0733 1 then the remainder of the name is put in the HDR4 label with the
353 M 0734 1 size that is in the HDR4 label only.
354 M 0735 1 PLEASE NOTE that the actual implementation of this is different for
355 M 0736 1 volumes with a 4 in the VOL1 standard field as opposed to a 3 or less.
356 M 0737 1 This is because the new standard allows us to write any kind
357 M 0738 1 of data in implementation dependant fields. The old standard did not allow
358 M 0739 1 us to do this.
359 M 0740 1
360 M 0741 1 BEGIN
361 M 0742 1 BIND
362 M 0743 1 CVT2 = DESCRIPTOR ('!2ZW');
363 M 0744 1 LOCAL
364 M 0745 1 DESCR : VECTOR [2, LONG],
365 M 0746 1 MVL : REF BBLOCK;
366 M 0747 1 MVL = .CURRENT_VCB[VCB$L_MVL];
367 M 0748 1 IF .FILE_SPEC_CEN LEQU HD1$S_FILEID
368 M 0749 1 THEN
369 M 0750 1 BEGIN
370 M 0751 1 IF .MVL[MVL$B_STDVER] GTR 3
371 M 0752 1 THEN
372 M 0753 1 HDR4[HD4$B_FILEID_EXT_SIZE] = 0
373 M 0754 1 ELSE
374 M 0755 1 CH$FILL('0', HD4$S_FILEID_EXT_V3, HDR4[HD4$T_FILEID_EXT_V3]);
375 M 0756 1 END
```

```
: 376      M 0757 1
: 377      M 0758 1
: 378      M 0759 1
: 379      M 0760 1
: 380      M 0761 1
: 381      M 0762 1
: 382      M 0763 1
: 383      M 0764 1
: 384      M 0765 1
: 385      M 0766 1
: 386      M 0767 1
: 387      M 0768 1
: 388      M 0769 1
: 389      M 0770 1
: 390      M 0771 1
: 391      M 0772 1
: 392      M 0773 1
: 393      M 0774 1
: 394      M 0775 1
: 395      M 0776 1
: 396      M 0777 1
: 397      M 0778 1
: 398      M 0779 1
: 399      M 0780 1
: 400      M 0781 1
: 401      M 0782 1
: 402      M 0783 1
: 403      M 0784 1
: 404      M 0785 1
: 405      M 0786 1
: 406      M 0787 1
: 407      M 0788 1
: 408      M 0789 1
: 409      M 0790 1
: 410      M 0791 1
: 411      M 0792 1

      ELSE
      BEGIN
        IF .MVL[MVL$B_STDVER] GTR 3
        THEN
          HDR4[HD4$B_FILEID_EXT_SIZE] = .FILE_SPEC_LEN - HD1$S_FILEID
        ELSE
          BEGIN
            LOCAL LEN;
            LEN = .FILE_SPEC_LEN - HD1$S_FILEID;
            DESCRI[0] = HD4$S_FILEID_EXT_V3;
            DESCRI[1] = HDR4[AD4$T_FILEID_EXT_V3];
            SFAO(CVT2,0,DESCR,.LEN);
          END;
        END;
      END;

      ! if enter function return file name string to user
      IF .LOCAL_FIB [ FIBSW_DID_NUM ] NEQ 0
      THEN
        RESULTANT_STRING ( .ANSI_NAME SIZE LSS 0,
                            FILE_SPEC MAX,
                            FILE_ID[0],
                            .VERSION );

      ! transform the VMS file version number into ANSI format
      CALC_TAPE_VER ( .VERSION, GEN_NUM_VER );
      DESCRIPT [ 0 ] = HD1$S_GENNO # HDT$S_GENVER;
      DESCRIPT [ 1 ] = HDR1 [ HD1$T_GENNO ];
      SYSSFAO ( DESCRIPTOR ('!4ZL!2ZL'), 0, DESCRIPT,
                .GEN_NUM_VER [ 0 ], .GEN_NUM_VER [ 1 ] );

      END;
    %: ! end of macro FORMAT_FILE_NAME
```

413 0793 1 GLOBAL ROUTINE FORMAT_HDRS : COMMON_CALL NOVALUE =
414 0794 1
415 0795 1 ++
416 0796 1
417 0797 1 FUNCTIONAL DESCRIPTION:
418 0798 1 This routine formats HDR1, HDR2, HDR3 and HDR4.
419 0799 1
420 0800 1 CALLING SEQUENCE:
421 0801 1 FORMAT_HDRS()
422 0802 1
423 0803 1 INPUT PARAMETERS:
424 0804 1 none
425 0805 1
426 0806 1 IMPLICIT INPUTS:
427 0807 1 CURRENT_VCB - address of current vcb
428 0808 1 HDR1 - address of HDR1 label
429 0809 1 HDR2 - address of HDR2 label
430 0810 1 HDR3 - address of HDR3 label
431 0811 1 HDR4 - address of HDR4 label
432 0812 1 LOCAL_FIB - copy of user's fib
433 0813 1
434 0814 1 OUTPUT PARAMETERS:
435 0815 1 none
436 0816 1
437 0817 1 IMPLICIT OUTPUTS:
438 0818 1 HDR1, HDR2, HDR3, and HDR4 formatted
439 0819 1
440 0820 1 ROUTINE VALUE:
441 0821 1 none
442 0822 1
443 0823 1 SIDE EFFECTS:
444 0824 1 none
445 0825 1
446 0826 1 --
447 0827 1
448 0828 2 BEGIN
449 0829 2
450 0830 2
451 0831 2 EXTERNAL ROUTINE CONVDATE_R2J. ! convert regular date to
452 0832 2 Julian for tape
453 0833 2 SYSSASCTIM : ADDRESSING_MODE(ABSOLUTE),! get ASCII date/time
454 0834 2 SYSSFAD : ADDRESSING_MODE(ABSOLUTE),! format ASCII output
455 0835 2 WRITE_ATTRIBUTE : COMMON_CALL; ! write user supplied attrbts
456 0836 2
457 0837 2
458 0838 2 EXTERNAL REGISTER
459 0839 2 COMMON_REG;
460 0840 2
461 0841 2
462 0842 2 EXTERNAL
463 0843 2 CURRENT_UCB : REF BBLOCK, ! address of current UCB
464 0844 2 HDR1 : REF BBLOCK, ! address of HDR1(EOF1) label
465 0845 2 HDR2 : REF BBLOCK, ! address of HDR2(EOF2) label
466 0846 2 HDR3 : REF BBLOCK, ! address of HDR3(EOF3) label
467 0847 2 IO_PACKET : REF BBLOCK, ! address of IO request packet
468 0848 2 LOCAL_FIB : BBLOCK; ! copy of user file info block
469 0849 2

```
: 470      0850 2      BIND
: 471      0851 2      CVT4      = DESCRIPTOR ( '14XW' );
: 472      0852 2      CVT5      = DESCRIPTOR ( '15ZW' );
: 473      0853 2      DEFAULT   = UPLIT ('00512');
: 474      0854 2      DEF HEX   = UPLIT ('0200');
: 475      0855 2      STARID   = UPLIT ('DECFILE11A');

: 476      0856 2
: 477      0857 2      GLOBAL
: 478      0858 2      NMBLOCK : BBLOCK [10];           ! name block
: 479      0859 2
: 480      0860 2      LOCAL
: 481      0861 2      DESCRIPTOR
: 482      0862 2      MVL       : REF BBLOCK;
: 483      0863 2      TODAY    : VECTOR [12, BYTE];        ! general Pdescriptor
: 484      0864 2
: 485      0865 2
: 486      0866 2      ! Blank fill the headers and default the fields
: 487      0867 2      ! in HDR1, HDR2, HDR3, and HDR4
: 488      0868 2
: 489      0869 2      CH$FILL(' ', 320, .HDR1);
: 490      0870 2      HDR1[HD1$L HD1$LID] = 'HDR1';
: 491      0871 2      MVL = .CURRENT_VCB[VCBSL_MVL];
: 492      0872 2      CH$MOVE(HD1$S FILESETID, MVL[MVL$T_SET_ID], HDR1[HD1$T_FILESETID]);
: 493      0873 2      CH$FILL('0', AD1$S_BLOCKCNT, HDR1[AD1$T_BLOCKCNT]);
: 494      0874 2
: 495      0875 2      ! If volume is for interchange do not write any VMS specific fields.
: 496      0876 2
: 497      0877 2      IF NOT .CURRENT_VCB[VCBSV_INTCHG]
: 498      0878 2      THEN
: 499      0879 2      CH$MOVE(10, STARID, HDR1[HD1$T_SYSODE]);
: 500      0880 2
: 501      0881 2      ! default expiration and creation date
: 502      0882 2
: 503      0883 2      DESCRIPTOR[0] = 12;
: 504      0884 2      DESCRIPTOR[1] = TODAY;
: 505      0885 2      SY$ASCTIM(0, DESCRIPTOR, 0, 0);          ! get today's date in dd-mmm-yyyy
: 506      0886 2      CONVDATE_R2J(TODAY, HDR1[HD1$T_CREATEDT]);
: 507      0887 2      CH$MOVE(AD1$S_CREATEDT, HDR1[HD1$T_CREATEDT], HDR1[HD1$T_EXPIREDT]);
: 508      0888 2
: 509      0889 2      ++
: 510      0890 2
: 511      0891 2      format HDR2 defaults
: 512      0892 2
: 513      0893 2
: 514      0894 2      HDR2[HD2$L HD2$LID] = 'HDR2';
: 515      0895 2      HDR2[HD2$S$RECFORMAT] = 'F';
: 516      0896 2      DESCRIPTOR[0] = HD2$S_BLOCKLEN;
: 517      0897 2      DESCRIPTOR[1] = HDR2[AD2$T_BLOCKLEN];
: 518      0898 2
: 519      0899 2      ! fill in the blocksize
: 520      0900 2
: 521      0901 2      IF NOT SFAD(CVT5, 0, DESCRIPTOR, .CURRENT_UCB[UCBSU_DEVBUFSIZE])
: 522      0902 2      THEN CH$MOVE(HD2$S_BLOCKLEN, DEFAULT, HDR2[HD2$T_BLOCKLEN]);
: 523      0903 2
: 524      0904 2      ! default the record size to be the blocksize
: 525      0905 2
: 526      0906 2      CH$MOVE(HD2$S_RECLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]);
```

527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583

0907 2
0908 2
0909 2
0910 2
0911 2
0912 2
0913 3
0914 3
0915 3
0916 4
0917 3
0918 3
0919 2
0920 2
0921 2
0922 2
0923 2
0924 2
0925 2
0926 2
0927 2
0928 2
0929 2
0930 2
0931 2
0932 2
0933 2
0934 2
0935 2
0936 2
0937 2
0938 2
0939 2
0940 2
0941 2
0942 2
0943 2
0944 2
0945 2
0946 2
0947 2
0948 2
0949 2
0950 2
0951 2
0952 2
0953 2
0954 2
0955 2
0956 2
0957 3
0958 4
0959 3
0960 4
0961 5
0962 4
0963 3

| use the record size if it exists
| IF .CURRENT_VCB[VCBSW_RECORDSZ] NEQ 0
| THEN
| BEGIN
| DESC[0] = HD2SS_RECLEN;
| DESC[1] = HDR2[HD2ST_RECLEN];
| IF NOT \$FAO(CVT5, 0, DESC, .CURRENT_VCB[VCBSW_RECORDSZ])
| THEN
| CHSMOVE(HD2SS_RECLEN, HDR2[HD2ST_BLOCKLEN], HDR2[HD2ST_RECLEN]);
| END;
|
| HDR2[HD2ST_BUFOFF] = '00';
|
| ++
| default HDR3 (sequencial files, fixed lenght block size)
|
| --
| HDR3[HD3SL_HD3LID] = 'HDR3';
| CHSFILL('0', HD3SS_RECATTR, HDR3[HD3ST_RECATTR]);
| (HDR3[HD3ST_RECATTR] + 4)<0, 32> = '0201';
|
| ++
| default HDR4 no long file name, make the default dependant on the
| ANSI version type
|
| --
| HDR4[HD4SL_HD4LID] = 'HDR4';
| IF .MVL[MV[SB_STDVER] GTR 3]
| THEN
| HDR4[HD4SB_FILEID_EXT_SIZE] = 0
| ELSE
| CHSFILL('0', HD4SS_FILEID_EXT_V3, HDR4[HD4ST_FILEID_EXT_V3]);
|
| ++
| fill in the RMS default record size
| if record size on mount then use it
| else if blocks size the use it
| else default
|
| --
| DESC[0] = 4;
| DESC[1] = HDR3[HD3ST_RECATTR];
| IF .CURRENT_VCB[VCBSW_RECORDSZ] NEQ 0
| THEN
| BEGIN
| IF NOT \$FAO(CVT4, 0, DESC, .CURRENT_VCB[VCBSW_RECORDSZ])
| THEN
| BEGIN
| IF NOT \$FAO(CVT4, 0, DESC, .CURRENT_UCB[UCBSW_DEVBUFSIZ])
| THEN CHSMOVE (4, DEF_HEX, HDR3[HD3ST_RECATTR]);
| END;

```

584      0964 3      END
585      0965 3      ELSE
586      0966 4      BEGIN
587      0967 4      IF NOT SFAD( CVT4, 0, DESCR, CURRENT_UCB[UCBSW_DEVBUFSIZ] )
588      0968 3      THEN CHSMOVE ( 4, DEF_HEX, HDR3[HDSST_RECATTR] );
589      0969 2      END;
590      0970 2
591      0971 2
592      0972 2      ! pickup user supplied attributes
593      0973 2
594      0974 2      CHSFILL ( 0, 10, NMBLOCK );
595      0975 2      WRITE_ATTRIBUTE ();
596      0976 2
597      0977 2      !
598      0978 2
599      0979 2      set up the file specification
600      0980 2
601      0981 2
602      0982 2      FORMAT_FILE_NAME;
603      0983 2
604      0984 1      END:           ! end of routine FORMAT_HDRS

```

```

.TITLE FRMHDR
.IDENT \V04-000\

.PSECT SCODES,NOWRT,2
      57 58 34 21 00000 P.AAB: .ASCII \!4XW\
      00000004 00004 P.AAA: .LONG 4
      00000000 00008 P.AAD: .ADDRESS P.AAB
      57 5A 35 21 0000C P.AAC: .ASCII \!5ZW\
      00000004 00010 P.AAE: .LONG 4
      00000000 00014 P.AAF: .ADDRESS P.AAD
      00 00 00 32 31 35 30 30 00018 P.AAG: .ASCII \00512\<0><0><0>
      30 30 32 30 00020 P.AAI: .ASCII \0200\
      00 00 41 31 31 45 4C 49 46 43 45 44 00024 P.AAH: .ASCII \DECFILE11A\<0><0>
      57 5A 32 21 00030 P.AAK: .ASCII \!2ZW\
      00000004 00034 P.AAJ: .LONG 4
      00000000 00038 P.AAL: .ADDRESS P.AAI
      4C 5A 32 21 4C 5A 34 21 0003C P.AAL: .ASCII \!4ZL!2ZL\
      00000008 00044 P.AAM: .LONG 8
      00000000 00048 P.AAN: .ADDRESS P.AAK

.PSECT SLOCKEDD1$,NOEXE,2
      00000 NMBLOCK:: .BLKB 10
      CVT4= P.AAA
      CVT5= P.AAC
      DEFAULT= P.AAE
      DEF HEX= P.AAF
      STARID= P.AAG
      CVT2= P.AAH
      .EXTRN CONVDATE_R2J, SYSSASCTIM
      .EXTRN SYSSFAO, WRITE_ATTRIBUTE

```

				07FC 00000		
0140	8F	20				
06	15	A6	OC	SA 00000000G	00 9E 00002	.EXTRN CURRENT UCB, HDR1
				SE FF24 0000G	CE 9E 00009	
	30	A6	A7	56 0000G	CF D0 0000E	.EXTRN IO PACKET, LOCAL FIB
				6E	00 2C 00013	.EXTRN CALC TAPE VER, PARSE_NAME_TYPE
	3C	06	AB	66 31524448	66 0001A	.EXTRN PARSE_QUOTED_NAME
				57 34	8F D0 0001B	.EXTRN RESULTANT_STRING
	A6	36	6E	AB D0 00022	06 28 00026	.EXTRN STRIP VERSION, ANSI_NAME_SIZE
					00 2C 0002C	WORK_AREA
	A6	2C	AB	04 E0 00033		.PSECT \$CODES,NOWRT,2
				9C AF	0A 28 00038	
	3C	A6	F8 AD	0C D0 0003E	18:	.ENTRY FORMAT_HDRS, Save R2,R3,R4,R5,R6,R7,R8,R9,- : 0793
				FC AD	AD 9E 00042	R10
	7E	00000000G	9F CF	04 FB 0004E		MOVAB SYSSFAO, R10
				0000G	29 C1 00055	-220(SP), SP
	2F	A0	29	EC AD 9F 0005B		MOVL HDR1, R6
				0000G	02 FB 0005E	MOVCS #0, (SP), #32, #320, (R6)
	A0	50	A0	50 0000G	06 28 00063	BBS #8, 44(CURRENT VCB), 18
				50 0000G	CF D0 0006E	MOVCS #10, STARID, 60(R6)
	F8	60	32524448	60 32524448	06 28 00073	MOVAB #12, DESCRIPTOR
				A0 46	BF D0 00073	MOVL TODAY, DESCRIPTOR+4
	FC	A0	46	04 AD 90 0007A		CLRL -(SP)
				AD 05	05 D0 0007F	PUSHAB DESCRIPTOR
	A0	50	50	05 A0 9E 00083		CLRL -(SP)
				7E 42	50 D0 00088	CALLS #4, #SYSSASCTIM
	A0	7E	F8	50 A0 9C 0008D		ADDL3 #41, HDR1, -(SP)
				AD 7E	AD 9F 00091	PUSHAB TODAY
	05	A0	FF2A	50 AD 9F 00094		CALLS #2, CONVDATE_R2J
				OC 04	CF 9F 00096	MOVL HDR1, R0
	A0	50	OC	50 OC 0009A		MOVL HDR2, R0
				50 04	50 E8 0009D	MOVL #844252232, (R0)
	05	A0	50	50 0000G		MOVB #70, 4(R0)
				50 0000G	CF D0 000A0	MOVL #5, DESCRIPTOR
	A6	56	56	56 0000G	05 28 000A5	MOVAB 5(R0), DESCRIPTOR+4
				58 05	CF D0 000AC	MOVL CURRENT_UCB, R0
	0A	A6	05	58 05	05 28 000B1	MOVL 66(R0), -(SP)
				58 05	AB 3C 0C0B7	PUSHAB DESCRIPTOR
	A6	58	58	58 59	59 D4 000BB	CLRL -(SP)
				58 59	D5 000BD	PUSHAB CVT5
	0A	58	59	59 27	59 13 000BF	CALLS #4, SYSSFAO
				59 27	D6 000C1	BLBS R0, 28
	A6	05	58	59 59	59 D6 000C1	MOVL HDR2, R0
				59 59	D6 000C1	MOVC3 #5, DEFAULT, 5(R0)
	0A	A6	58	58 59	59 D6 000C1	MOVL HDR2, R6
				58 59	D5 000BD	MOVC3 #5, 10(R6)
	A6	05	58	58 27	59 13 000BF	MOVZWL 80(CURRENT_VCB), R8
				58 27	13 000BF	TSTL R8
	0A	A6	58	58 59	59 D6 000C1	BEQL 3\$
				58 59	D6 000C1	INCL R9

		F8	AD	05	00	000C3	MOVL	#5, DESCRIPTOR	0913	
		FC	AD	0A	A6	9E 000C7	MOVAB	10(R6), DESCRIPTOR+4	0914	
				F8	58	DD 000CC	PUSHL	R8	0916	
					AD	9F 000CE	PUSHAB	DESCRIPTOR		
					7E	D4 000D1	CLRL	-(SP)		
				FEED	CF	9F 000D3	PUSHAB	CVT5		
				6A	04	FB 000D7	CALLS	#4, SYSSFAO		
				0B	50	E8 000DA	BLBS	R0, 38		
				05	A0	0000G	MOVL	HDR2, R0	0918	
				50	50	0000G	MOVC3	#5(R0), 10(R0)	0921	
				32	A0	3030	MOVL	HDR2, R0		
				56	56	0000G	MOVW	#12336, 50(R0)	0928	
				66	66	33524448	MOVL	HDR3, R6		
0040	8F	30	6E	00	2C	000FF	MOVCS	#861029448, (R6)	0929	
				04	A6	00106		#0, (SP), #48, #64, 4(R6)		
				08	A6	31303230	MOVL	#825242160, 8(R6)	0930	
				50	50	0000G	MOVL	HDR4, R0	0938	
				60	60	34524448	MOVL	#877806664, (R0)		
				03	22	A7 91 0011C	CMPB	34(MVL), #3	0939	
					05	18 00120	BLEQU	4S		
					04	A0 94 00122	CLRB	4(R0)	0941	
				43	A0	3030	BRB	5S		
				F8	04	8F 00127	MOVW	#12336, 67(R0)	0943	
				FC	AD	D0 0012D	MOVL	#4, DESCRIPTOR	0953	
					04	A6 9E 00131	MOVAB	4(R6), DESCRIPTOR+4	0954	
					11	59 E9 00136	BLBC	R9, 6\$	0955	
					58	DD 00139	PUSHL	R8	0958	
					F8	AD 9F 0013B	PUSHAB	DESCRIPTOR		
						7E D4 0013E	CLRL	-(SP)		
					FE74	CF 9F 00140	PUSHAB	CVT4		
				6A	04	FB 00144	CALLS	#4, SYSSFAO		
				23	50	E8 00147	BLBS	R0, 78		
				50	7E	0000G	MOVL	CURRENT_UCB, R0	0967	
					42	A0 3C 0014F	MOVZL	66(R0), -(SP)		
					F8	AD 9F 00153	PUSHAB	DESCRIPTOR		
						7E D4 00156	CLRL	-(SP)		
					FE5C	CF 9F 00158	PUSHAB	CVT4		
				6A	04	FB 0015C	CALLS	#4, SYSSFAO		
				0B	50	E8 0015F	BLBS	R0, 78		
				50	0000G	MOVL	HDR3, R0	0968		
				04	A0	FE69	MOVL	DEF_HEX, 4(R0)	0974	
				6E	6E	00	MOVCS	#0, -(SP), #0, #10, NMBLOCK		
					0000'	2C	0016D	78:		
					0000G	CF	00172			
						00	FB 00175	CALLS	#0, WRITE_ATTRIBUTE	0975
						0000G	CF DO 0017A	MOVL	IO_PACKET, R0	
						2C	BO 0017F	MOVL	34(R0), ABD	
						12	A0 B5 00183	TSTW	18(ABD)	
						03	13 00186	BEQL	8\$	
						0099	31 00188	BRW	19\$	
						OC	AE D4 0018B	CLRL	FILE_SPEC_LEN	
							54 D4 0018E	CLRL	K	
						04	54 E9 00190	BLBC	10\$	
							55 D4 00193	CLRL	R5	
							03 11 00195	BRB	11\$	
						55	02 00 00197	MOVL	#2, R5	
						50	01 CE 0019A	MNEG	#1, I	
							10\$:			
							11\$:			

51	50	52	0000GCF	58	11	0019D	12\$:	BRB	17\$	K, I, R1	NMBLOCK[R1], R2	#1600, R2, R1	R1, CHARS
51	52	00000640		54	C1	0019F		ADDL3					
51	6E			41	3C	001A3		MOVZWL					
00	52			8F	C7	001A9		DIVL3					
51	51			51	90	001B1		MOVB					
00	8E			28	C7	001B4		DIVL3					
51	52			01	7A	001B8		EMUL					
01	AE			2B	78	001BD		EDIV					
00	52			51	90	001C2		MOVAB					
51	8E			01	7A	001C6		EMUL					
02	AE			2B	7B	001CB		EDIV					
				51	90	001D0		MOVAB					
				51	D4	001D4		CLRL					
				53	6E41	9A	001D6	13\$:	MOVZBL				
					1A	13	001DA		BEQL				
				52	70	AE	001DC		MOVAB				
				1E		53	91	CMPB					
						91	001E0		NAME STRING, R2				
						06	1E	BGEQU					
				53	40	A5	001E5		MOVAB				
						D3	11	MOVAB					
						12	C0	BRB					
				53		90	001EB	14\$:	ADDL2				
						90	001EE	15\$:	MOVAB				
				OC	BE42	AE	D6	INCL					
						D3	001F3		AOBLEQ				
						F3	001F6	16\$:					
						F3	001FA	17\$:	AOBLEQ				
						54	D5	TSTL					
						D5	001FE						
						OC	12	BNEQ					
						OC	00200		NAME_STRING, R0				
						70	AE	MOVAB					
				50	OC BE40	2E	9E	MOVAB					
						90	00202	14\$:					
						OC	00206						
						AE	D6	INCL					
						D6	0020B		ACBL				
						03	F1	MOVW					
						CF	B0	MOVAB					
						90	00214	15\$:					
						70	AE	MOVAB					
						9E	00219						
						04	01	BICB2					
						BA	0021E						
						25	11	BRB					
						A0	3C	MOVZWL					
						12	00224	16\$:					
						10	A0	MOVAB					
						9E	00229						
						51	61	MOVZWL					
						50	3C	MOVAB					
						01	A140	MOVAB					
						04	9E	MOVAB					
						AE	00230	PUSHAB					
						7E	D4	CLRL					
						10	AE	PUSHAB					
						18	AE	PUSHAB					
						04	FB	CALLS					
				0000G	CF	50	B0	MOVW					
					57	B1	00246	CMPW					
					8F	57	B1	VERSION					
						04	1B	VERISON,					
						0820	8F	#32767					
						00	BF	21\$:					
						20	00	BLEQU					
						0000G	2C	CHMU					
						CF	95	MOVCS					
						3A	00250						
						18	00261	TSTB					
						AE	9A	BGEQ					
						4F	00263	ANSI_NAME_SIZE					
						20	AE	MOVZBL					
						04	9E	FILE_ID, DESCRIPT					
						E4	AE	DESCRIPT					
						E8	AD	FILÉ_ID, DESCRIPT+4					
						15	20	MOVAB					
						04	AE	BLBC					
						E4	AD	PUSHAB					
						0C	AE	PUSHL					
						DD	00274	DESCRIPT					
								FILE_SPEC_PTR					

			14	AE DD 00277	PUSHL	FILE_SPEC_LEN
			05	FB 0027A	CALLS	#3, PARSE_QUOTED_NAME
			01	90 0027F	MOVB	#1 ANSI_NAME_SIZE
			17	11 00284	BRB	23\$
			E4	AD 9F 00286	PUSHAB	DESCRIPT
			OC	DD 00289	PUSHL	FILE_SPEC_PTR
			14	DD 0028C	PUSHL	FILE_SPEC_LEN
			7E	D4 0028F	CLRL	-(SP)
			04	FB 00291	CALLS	#4, PARSE_NAME_TYPE
			50	E8 00296	BLBS	R0, 23\$
			0818	BF 00299	CHMU	#2072
			0000G	CF 95 0029D	TSTB	ANSI_NAME_SIZE
			24	15 002A1	BLEQ	24\$
			E4	8F 9A 002A3	MOVZBL	#79, DESCRIPT
			E8	AD 9E 002A8	MOVAB	WORK_AREA, DESCRIPT+4
			E4	AD 9F 002AE	PUSHAB	DESCRIPT
			24	AE 9F 002B1	PUSHAB	FILE_ID
			7E	4F 9A 002B4	MOVZBL	#79, -(SP)
				7E D4 002B8	CLRL	-(SP)
			04	FB 002BA	CALLS	#4, PARSE_NAME_TYPE
			50	E9 002BF	BLBC	R0, 24\$
			05	01 8E 002C2	MNEG8	#1, ANSI_NAME_SIZE
			0000G	CF D0 002C7	MOVL	HDR1, R0
04	A0	20	AE	11 28 002CC	MOVC3	#17, FILE_ID, 4(R0)
			56	0000G CF D0 002D2	MOVL	HDR4, R6
05	A6	31	AE	3E 28 002D7	MOVC3	#62, FILE_ID+17, 5(R6)
			50	AB D0 002DD	MOVL	52(CURRENT_VCB), MVL
			11	OC AE D1 002E1	CMPL	FILE_SPECLEN, #17
			03	13 1A 002E5	BGTRU	26\$
			22	A0 91 002E7	CMPB	34(MVL), #3
			05	1B 002EB	BLEQU	25\$
			04	A6 94 002ED	CLRB	4(R6)
			43	33 11 002F0	BRB	28\$
			A6	3030 8F B0 002F2	MOVW	#12336, 67(R6)
				2B 11 002F8	BRB	28\$
51	OC	AE	03	11 C3 002FA	SUBL3	#17, FILE_SPEC_LEN, R1
			22	A0 91 002FF	CMPB	34(MVL), #3
			04	1B 00303	BLEQU	27\$
			A6	51 90 00305	MOVB	R1, 4(R6)
				1A 11 00309	BRB	28\$
			10	51 D0 0030B	MOVL	R1, LEN
			14	02 D0 0030E	MOVL	#2, DESCR
			A6	9E 00312	MOVAB	67(R6), DESCR+4
			43	50 DD 00317	PUSHL	LEN
				14 AE 9F 00319	PUSHAB	DESCR
			7E	7E D4 0031C	CLRL	-(SP)
			FCC6	CF 9F 0031E	PUSHAB	CVT2
			6A	04 FB 00322	CALLS	#4, SYSSFAO
				0000G CF B5 00325	TSTW	LOCAL_FIB+10
			19	13 00329	BEQL	30\$
			7E	57 3C 0032B	MOVZWL	VERSION, -(SP)
			24	AE 9F 0032E	PUSHAB	FILE_ID
			7E	8F 9A 00331	MOVZBL	#79, -(SP)
				7E D4 00335	CLRL	-(SP)
			0000G	CF 95 00337	TSTB	ANSI_NAME_SIZE
				02 18 0033B	BGEQ	29\$
			6E	D6 0033D	INCL	(SP)

0000G	CF		04	FB 0033F	29\$: CALLS #4, RESULTANT_STRING
		18	AE 9F 00344	30\$: PUSHAB GEN_NUM_VER	
0000G	7E		57 3C 00347	MOVZWL VERSION_--(SP)	
	CF		02 FB 0034A	CALLS #2, CALC_TAPE_VER	
E8 AD	E4 AD		06 D0 0034F	MOVL #6, DESCRIPT	
	0000G CF		23 C1 00353	ADDL3 #35, HDR1, DESCRIPT+4	
		1C AE DD 0035A	PUSHL GEN_NUM_VER+4		
		1C AE DD 0035D	PUSHL GEN_NUM_VER		
		E4 AD 9F 00360	PUSHAB DESCRIPT		
		7E D4 00363	CLRL -(SP)		
		FC8F CF 9F 00365	PUSHAB P_AAJ		
		6A 05 FB 00369	CALLS #5, SYSSFAO		
			04 0036C	RET	

0984

; Routine Size: 877 bytes, Routine Base: \$CODE\$ + 004C

605	0985 1
606	0986 1 END
607	0987 1
608	0988 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	953	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$LOCKEDD1\$	10	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----	Pages Mapped	Processing Time
	Total Loaded Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619 50 0	1000	00:01.9

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:FRMHDR/OBJ=OBJ\$:FRMHDR MSRC\$:FRMHDR/UPDATE=(ENH\$:FRMHDR)

609	0989 0
Size:	877 code + 86 data bytes
Run Time:	00:19.9

FRMHDR
V04-000

E 13
16-Sep-1984 02:19:38 VAX-11 Bliss-32 V4.0-742

Page 18

: Elapsed Time: 00:38.7
: Lines/CPU Min: 2989
: Lexemes/CPU-Min: 23271
: Memory Used: 299 pages
: Compilation Complete

FRM
V04

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

